We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

3

Basic Extended Kalman Filter – Simultaneous Localisation and Mapping

[1] Oduetse Matsebe, [2] Molaletsa Namoshe and [3] Nkgatho Tlale
[1,2,3] *Council for Scientific and Industrial Research, Mechatronics and Micro Manufacturing Pretoria, South Africa*

1. Introduction

Most real systems are non-linear. Extended Kalman Filter (EKF) uses non-linear models of both the process and observation models while the Kalman Filter (KF) uses linear models. EKF is a good way to learn about Simultaneous Localisation and Mapping (SLAM). Much of the literature concentrates on advanced SLAM methods which stems from EKF or uses probabilistic techniques. This makes it difficult for new researchers to understand the basics of this exciting area of research.

SLAM asks if it is possible for a robot, starting with no prior information, to move through its environment and build a consistent map of the entire environment. Additionally, the vehicle must be able to use the map to navigate and hence plan and control its trajectory during the mapping process. The applications of a robot capable of navigating, with no prior map, are diverse indeed. Domains in which 'man in the loop' systems are impractical or difficult such as sub-sea surveys and disaster zones are obvious candidates. Beyond these, the sheer increase in autonomy that would result from reliable, robust navigation in large dynamic environments is simply enormous (Newman 2006). SLAM has been implemented in a number of different domains from indoor robots to outdoor, underwater, and airborne systems. In many applications the environment is unknown. A priori maps are usually costly to obtain, inaccurate, incomplete, and become outdated. It also means that the robot's operation is limited to a particular environment (Neira 2008).

This goal of the chapter is to provide an opportunity for researchers who are new to, or interested in, this exciting area with the basics, background information, major issues, and the state-of-the-art as well as future challenges in SLAM with a bent towards EKF-SLAM. It will also be helpful in realizing what methods are being employed and what sensors are being used. It presents the 2 – Dimensional (2D) feature based EKF-SLAM technique used for generating robot pose estimates together with positions of features in the robot's operating environment, it also highlights some of the basics for successful EKF – SLAM implementation: (1) Process and observation models, these are the underlying models required, (2) EKF-SLAM Steps, the three-stage recursive EKF-SLAM process comprising prediction, observation and update, (3) Feature Extraction and Environment modelling, a

process of extracting well defined entities or landmarks (features) which are recognisable and can be repeatedly detected to aid navigation, (4) Data Association, this consists of determining the origin of each measurement, in terms of map features, (5) Multi – Robot – EKF – SLAM, the two types of multi robot systems are described: Collaborative and Cooperative multi robot systems with more emphasis on the Cooperative SLAM Scheme.

2. Basic Structure of EKF - SLAM

The EKF-SLAM process consists of a recursive, three-stage procedure comprising prediction, observation and update steps. The EKF estimates the pose of the robot made up of the position (x_r, y_r) and orientation ψ_r, together with the estimates of the positions of the N environmental features $\boldsymbol{x}_{f,i}$ where $i = 1....N$, using observations from a sensor onboard the robot (Williams et al. 2001). We will constrain ourselves to using the simplest feature model possible; a point feature such that the coordinates of the i^{th} feature in the global reference frame are given by:

$$\boldsymbol{x}_{f,i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

SLAM considers that all landmarks are stationary, hence the state transition model for the i^{th} feature is given by:

$$\boldsymbol{x}_{f,i}(k) = \boldsymbol{x}_{f,i}(k-1) = \boldsymbol{x}_{f,i} \tag{2}$$

It is important to note that the evolution model for features does have any uncertainty since the features are considered static.

2.1 Process Model

Implementation of EKF - SLAM requires that the underlying state and measurement models be developed. This section describes the process models necessary for this purpose.

2.1.1 Kinematic Model

Modeling of the kinematic states involves the study of the geometrical aspects of motion. The motion of a robot through the environment can be modeled through the following discrete time non-linear model:

$$\boldsymbol{X}_r(k) = \boldsymbol{f}(\boldsymbol{X}_r(k-1), \boldsymbol{u}(k), k) \tag{3}$$

Thus, $\boldsymbol{X}_r(k)$ is the state of the robot at time k, $\boldsymbol{u}(k)$ is the robot control input at time k. $\boldsymbol{f}(,,)$ is a function that relates the robot state at time $k-1$, known control inputs to the robot state at time k.

$$\boldsymbol{u}(k) = \begin{bmatrix} u_1 \\ . \\ . \\ u_N \end{bmatrix} \tag{4}$$

Equation (4) above is a little unrealistic, we need to model uncertainty. One popular way to model uncertainty is to insert noise terms into the control input $\boldsymbol{u}(k)$ such that:

$$\boldsymbol{u}(k) = \boldsymbol{u}_n(k) + \boldsymbol{\gamma}_u(k) \tag{5}$$

Thus $\boldsymbol{u}_n(k)$ is a nominal control input and $\boldsymbol{\gamma}_u(k)$ is a vector of control noise which is assumed to be temporally uncorrelated, zero mean and Gaussian with standard deviation σ.

$$\boldsymbol{\gamma}_u(k) = \begin{bmatrix} \sigma_1 \\ . \\ . \\ \sigma_N \end{bmatrix} \tag{6}$$

The strength (covariance) of the control noise is denoted $\boldsymbol{Q}_u$, and is given by:

$$\boldsymbol{Q}_u = diag\begin{pmatrix} \sigma_1^2 & . & . & \sigma_N^2 \end{pmatrix} \tag{7}$$

The complete discrete time non-linear kinematic model can now be expressed in general form as:

$$\boldsymbol{X}_r(k) = f(\boldsymbol{X}_r(k-1), \boldsymbol{u}_n(k) + \boldsymbol{\gamma}_u(k)) \tag{8}$$

2.1.2 Using Dead-Reckoned Odometry Measurements

Sometimes a navigation system will be given a dead reckoned odometry position as input without recourse to the control signals that were involved. The dead reckoned positions can be converted into a control input for use in the core navigation system. It would be a bad

idea to simply use a dead-reckoned odometry estimate as a direct measurement of state in a Kalman Filter (Newman 2006).

Given a sequence $\boldsymbol{x}_o(1), \boldsymbol{x}_o(2), \boldsymbol{x}_o(3)...\boldsymbol{x}_o(k)$ of dead reckoned positions, we need to figure out a way in which these positions could be used to form a control input into a navigation system. This is given by:

$$\boldsymbol{u}_o(k) = \Theta \boldsymbol{x}_o(k-1) \oplus \boldsymbol{x}_o(k) \tag{9}$$

This is equivalent to going back along $\boldsymbol{x}_o(k-1)$ and forward along $\boldsymbol{x}_o(k)$. This gives a small control vector $\boldsymbol{u}_o(k)$ derived from two successive dead reckoned poses. Equation (9) substracts out the common dead-reckoned gross error (Newman 2006). The plant model for a robot using a dead reckoned position as a control input is thus given by:

$$\boldsymbol{X}_r(k) = \boldsymbol{f}(\boldsymbol{X}_r(k-1), \boldsymbol{u}(k)) \tag{10}$$

$$\boldsymbol{X}_r(k) = \boldsymbol{X}_r(k-1) \oplus \boldsymbol{u}_o(k) \tag{11}$$

Θ and $\oplus$ are composition transformations which allows us to express robot pose described in one coordinate frame, in another alternative coordinate frame. These composition transformations are given below:

$$\boldsymbol{x}_1 \oplus \boldsymbol{x}_2 = \begin{bmatrix} x_1 + x_2 \cos\theta_1 - y_2 \sin\theta_1 \\ y_1 + x_2 \sin\theta_1 + y_2 \cos\theta_1 \\ \theta_1 + \theta_2 \end{bmatrix} \tag{12}$$

$$\Theta \boldsymbol{x}_1 = \begin{bmatrix} -x_1 \cos\theta_1 - y_1 \sin\theta_1 \\ x_1 \sin\theta_1 - y_1 \cos\theta_1 \\ -\theta_1 \end{bmatrix} \tag{13}$$

2.2 Measurement Model

This section describes a sensor model used together with the above process models for the implementation of EKF - SLAM. Assume that the robot is equipped with an external sensor capable of measuring the range and bearing to static features in the environment. The measurement model is thus given by:

$$\boldsymbol{z}(k) = \boldsymbol{h}(\boldsymbol{X}_r(k), x_i, y_i) + \boldsymbol{\gamma}_h(k) = \begin{bmatrix} r_i(k) \\ \theta_i(k) \end{bmatrix} \tag{14}$$

$$r_i = \sqrt{\left(x_i - x_r\right)^2 + \left(y_i - y_r\right)^2} \tag{15}$$

$$\theta_i = \tan^{-1}\left[\frac{y_i - y_r}{x_i - x_r}\right] - \psi_r \tag{16}$$

(x_i, y_i) are the coordinates of the i^{th} feature in the environment. $\boldsymbol{X}_r(k)$ is the (x, y) position of the robot at time k. $\boldsymbol{\gamma}_h(k)$ is the sensor noise assumed to be temporally uncorrelated, zero mean and Gaussian with standard deviation σ. $r_i(k)$ and $\theta_i(k)$ are the range and bearing respectively to the i^{th} feature in the environment relative to the vehicle pose.

$$\boldsymbol{\gamma}_h(k) = \begin{bmatrix} \sigma_r \\ \sigma_\theta \end{bmatrix} \tag{17}$$

The strength (covariance) of the observation noise is denoted $\boldsymbol{R}$.

$$\boldsymbol{R} = diag\begin{pmatrix} \sigma_r^2 & \sigma_\theta^2 \end{pmatrix} \tag{18}$$

2.3 EKF SLAM Steps

This section presents the three-stage recursive EKF-SLAM process comprising prediction, observation and update steps. Figure 1 below summarises the EKF - SLAM process described here.

2.3.1 Map Initialisation

The selection of a base reference B to initialise the stochastic map at time step 0 is important. One way is to select as base reference the robot's position at step 0. The advantage in choosing this base reference is that it permits initialising the map with perfect knowledge of the base location (Castellanos et al. 2006).

$$\boldsymbol{X}_0^B = \boldsymbol{X}_r^B = 0 \tag{19}$$

$$\boldsymbol{P}_0^B = \boldsymbol{P}_r^B = 0 \tag{20}$$

This avoids future states of the vehicle's uncertainty reaching values below its initial settings, since negative values make no sense. If at any time there is a need to compute the vehicle location or the map feature with respect to any other reference, the appropriate transformations can be applied. At any time, the map can also be transformed to use a feature as base reference, again using the appropriate transformations (Castellanos et al. 2006).

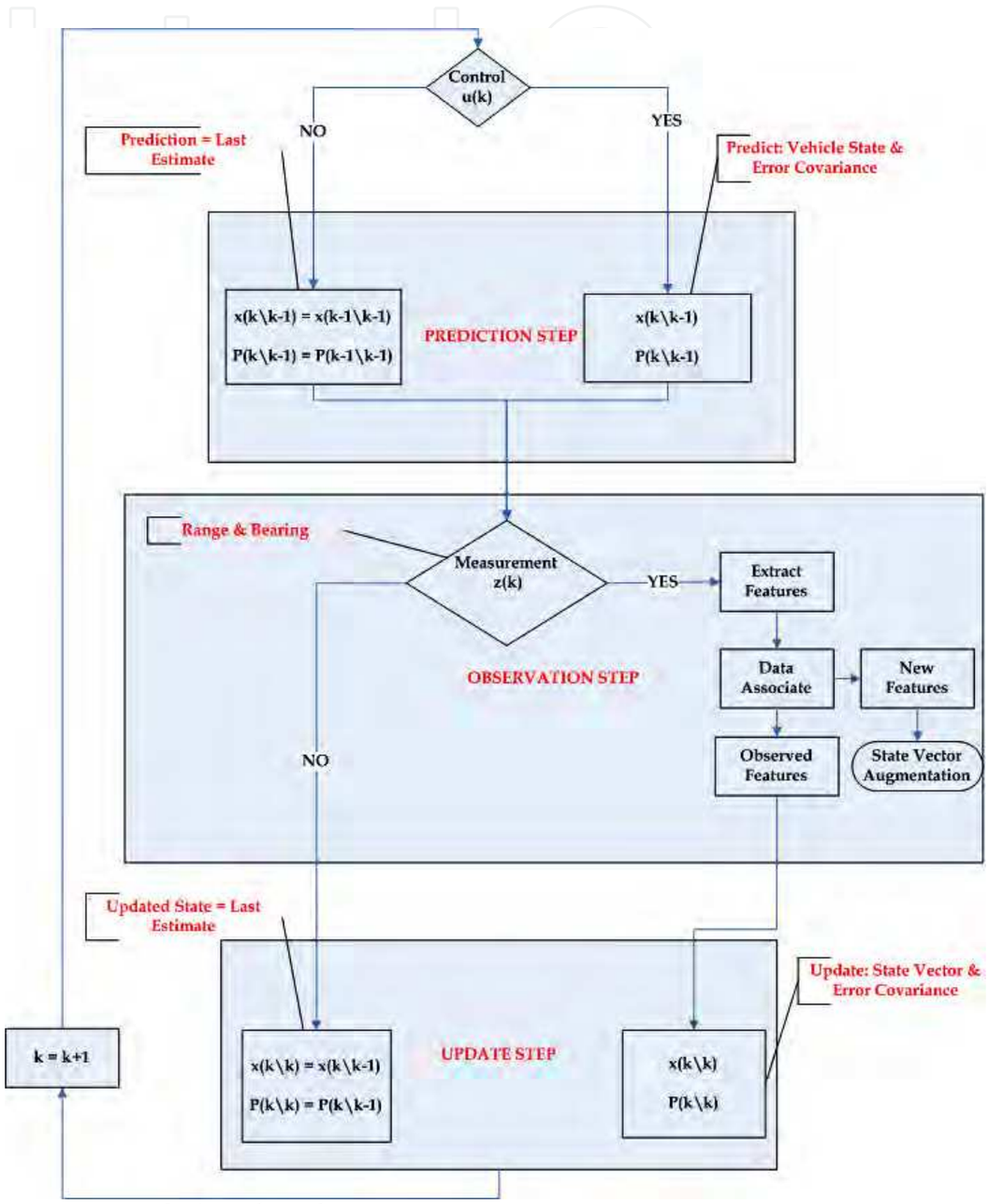


Fig. 1. Basic EKF-SLAM Flow chart

2.3.2 Prediction

(a) Prediction based on kinematic model

The prediction stage is achieved by passing the last estimate through the non-linear model of motion of the robot to compute the pose at instant k based on a control input $\boldsymbol{u}(k)$ and using the information up to instant $k-1$ (Williams et al. 2001). The predicted robot state $\boldsymbol{X}_r$ is thus given by:

$$\boldsymbol{X}_r(k \mid k-1) = \boldsymbol{f}(\boldsymbol{X}_r(k-1 \mid k-1), \boldsymbol{u}(k)) \tag{21}$$

Now we need to propagate the state error covariance. The covariance of the robot state, $\boldsymbol{P}_r(k \setminus k-1)$ is computed using the gradient, $\boldsymbol{F}_x(k)$ of the state propagation equation (8) with respect to the robot pose, the control noise covariance, $\boldsymbol{Q}_u$ and, the Jacobian, $\boldsymbol{J}_u$ of the state propagation equation (8) with respect to the control input $\boldsymbol{u}(k)$.

$$\boldsymbol{P}_r(k \mid k-1) = \boldsymbol{F}_x(k)\boldsymbol{P}_r(k-1 \mid k-1)\boldsymbol{F}_x{}^T(k) + \boldsymbol{J}_u(k)\boldsymbol{Q}_u(k)\boldsymbol{J}_u^T(k) \tag{22}$$

$$\boldsymbol{F}_x(k) = \frac{\partial f}{\partial X_r} \tag{23}$$

$$\boldsymbol{J}_u(k) = \frac{\partial f}{\partial u} \tag{24}$$

(b) Prediction using Dead-Reckoned Odometry Measurements as inputs

The prediction stage is achieved by a composition transformation of the last estimate with a small control vector calculated from two successive dead reckoned poses.

$$\boldsymbol{X}_r(k \mid k-1) = \boldsymbol{X}_r(k-1 \mid k-1) \oplus \boldsymbol{u}_o(k) \tag{25}$$

The state error covariance of the robot state, $\boldsymbol{P}_r(k \setminus k-1)$ is computed as follows:

$$\boldsymbol{P}_r(k \mid k-1) = \boldsymbol{J}_1(\boldsymbol{X}_r, \boldsymbol{u}_o)\boldsymbol{P}_r(k-1 \mid k-1)\boldsymbol{J}_1(\boldsymbol{X}_r, \boldsymbol{u}_o)^T + \boldsymbol{J}_2(\boldsymbol{X}_r, \boldsymbol{u}_o)\boldsymbol{U}_O(k)\boldsymbol{J}_1(\boldsymbol{X}_r, \boldsymbol{u}_o)^T \tag{26}$$

$\boldsymbol{J}_1(\boldsymbol{X}_r, \boldsymbol{u}_o)$ is the Jacobian of equation (11) with respect to the robot pose, $\boldsymbol{X}_r$ and $\boldsymbol{J}_2(\boldsymbol{X}_r, \boldsymbol{u}_o)$ is the Jacobian of equation (11) with respect to the control input, $\boldsymbol{u}_o$.

Based on equations (12), the above Jacobians are calculated as follows:

$$\boldsymbol{J}_1\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)=\frac{\partial\left(\boldsymbol{x}_1\oplus\boldsymbol{x}_2\right)}{\partial\boldsymbol{x}_1} \tag{27}$$

$$\boldsymbol{J}_1\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)=\begin{bmatrix}1 & 0 & -x_2\sin\theta_1-y_2\cos\theta_1\\ 0 & 1 & -x_2\cos\theta_1-y_2\sin\theta_1\\ 0 & 0 & 1\end{bmatrix} \tag{28}$$

$$\boldsymbol{J}_2\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)=\frac{\partial\left(\boldsymbol{x}_1\oplus\boldsymbol{x}_2\right)}{\partial\boldsymbol{x}_2} \tag{29}$$

$$J_2\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)=\begin{bmatrix}\cos\theta_1 & -\sin\theta_1 & 0\\ \sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix} \tag{30}$$

2.3.3 Observation

Assume that at a certain time k an onboard sensor makes measurements (range and bearing) to m features in the environment. This can be represented as:

$$\boldsymbol{z}_m(k)=[\boldsymbol{z}_1 \quad . \quad . \quad \boldsymbol{z}_m] \tag{31}$$

2.3.4 Update

The update step need not happen at every iteration of the filter. If at a given time step no observations are available then the best estimate at time k is simply the prediction $\boldsymbol{X}(k\mid k-1)$. If an observation is made of an existing feature in the map, the state estimate can now be updated using the optimal gain matrix $\boldsymbol{W}(k)$. This gain matrix provides a weighted sum of the prediction and observation. It is computed using the innovation covariance $\boldsymbol{S}(k)$, the state error covariance $\boldsymbol{P}(k\mid k-1)$ and the gradient of the observation model (equation 14), $\boldsymbol{H}(k)$.

$$\boldsymbol{W}(k)=\boldsymbol{P}(k\mid k-1)\boldsymbol{H}(k)\boldsymbol{S}^{-1}(k), \tag{32}$$

where $\boldsymbol{S}(k)$ is given by:

$$\boldsymbol{S}(k)=\boldsymbol{H}(k)\boldsymbol{P}(k\mid k-1)\boldsymbol{H}^{T}(k)+\boldsymbol{R}(k) \tag{33}$$

$\boldsymbol{R}(k)$ is the observation covariance.

This information is then used to compute the state update $\boldsymbol{X}(k \mid k)$ as well as the updated state error covariance $\boldsymbol{P}(k \mid k)$.

$$\boldsymbol{X}(k \mid k) = \boldsymbol{X}(k \mid k-1) + \boldsymbol{W}(k)\nu(k) \tag{34}$$

$$\boldsymbol{P}(k \mid k) = \boldsymbol{P}(k \mid k-1) - \boldsymbol{W}(k)\boldsymbol{S}(k)\boldsymbol{W}(k)^T \tag{35}$$

The innovation, $\boldsymbol{v}(k)$ is the discrepancy between the actual observation, $\boldsymbol{z}(k)$ and the predicted observation, $\boldsymbol{z}(k \mid k-1)$.

$$\boldsymbol{v}(k) = \boldsymbol{z}(k) - \boldsymbol{z}(k \mid k-1), \tag{36}$$

where $\boldsymbol{z}(k \mid k-1)$ is given as:

$$\boldsymbol{z}(k \mid k-1) = \boldsymbol{h}\left(\boldsymbol{X}_r(k \mid k-1), \boldsymbol{x}_i, \boldsymbol{y}_i\right) \tag{37}$$

$X_r(k \mid k-1)$ is the predicted pose of the robot and (x_i, y_i) is the position of the observed map feature.

2.4 Incorporating new features

Under SLAM the system detects new features at the beginning of the mission and when exploring new areas. Once these features become reliable and stable they are incorporated into the map becoming part of the state vector. A feature initialisation function $\boldsymbol{y}$ is used for this purpose. It takes the old state vector, $\boldsymbol{X}(k \mid k)$ and the observation to the new feature, $\boldsymbol{z}(k)$ as arguments and returns a new, longer state vector with the new feature at its end (Newman 2006).

$$\boldsymbol{X}(k \mid k)^* = \boldsymbol{y}\left[\boldsymbol{X}(k \mid k), \quad \boldsymbol{z}(k)\right] \tag{38}$$

$$\boldsymbol{X}(k \mid k)^* = \begin{bmatrix} \boldsymbol{X}(k \mid k) \\ x_r + r\cos(\theta + \psi_r) \\ y_r + r\sin(\theta + \psi_r) \end{bmatrix} \tag{39}$$

Where the coordinates of the new feature are given by the function $\boldsymbol{g}$:

$$\boldsymbol{g} = \begin{bmatrix} x_r + r\cos(\theta + \psi_r) \\ y_r + r\sin(\theta + \psi_r) \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \tag{40}$$

r and θ are the range and bearing to the new feature respectively. (x_r, y_r) and ψ_r are the estimated position and orientation of the robot at time k.
The augmented state vector containing both the state of the vehicle and the state of all feature locations is denoted:

$$\boldsymbol{X}(k \mid k)^* = [\boldsymbol{X}_r^T(k) \quad \boldsymbol{x}_{f,1}^T \quad . \quad . \quad \boldsymbol{x}_{f,N}^T] \tag{41}$$

We also need to transform the covariance matrix $\boldsymbol{P}$ when adding a new feature. The gradient for the new feature transformation is used for this purpose:

$$\boldsymbol{g} = \begin{bmatrix} x_r + r\cos(\theta + \psi_r) \\ y_r + r\sin(\theta + \psi_r) \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \tag{42}$$

The complete augmented state covariance matrix is then given by:

$$\boldsymbol{P}(k \mid k)^* = \boldsymbol{Y}_{x,z} \begin{bmatrix} \boldsymbol{P}(k \mid k) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \boldsymbol{Y}_{x,z}^T, \tag{43}$$

where $\boldsymbol{Y}_{x.z}$ is given by:

$$\boldsymbol{Y}_{x,z} = \begin{bmatrix} \boldsymbol{I}_{nxn} & \boldsymbol{0}_{nx2} \\ [\boldsymbol{G}_{x_r} \quad zeros(nstates - n)] & \boldsymbol{G}_z \end{bmatrix} \tag{44}$$

where $nstates$ and n are the lengths of the state and robot state vectors respectively.

$$\boldsymbol{G}_{X_r} = \frac{\partial g}{\partial X_r} \tag{45}$$

$$\boldsymbol{G}_{X_r} = \begin{bmatrix} \frac{\partial g_1}{\partial x_r} & \frac{\partial g_1}{\partial y_r} & \frac{\partial g_1}{\partial \psi_r} \\ \frac{\partial g_2}{\partial x_r} & \frac{\partial g_2}{\partial y_r} & \frac{\partial g_2}{\partial \psi_r} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -r\sin(\theta + \psi_r) \\ 0 & 1 & r\cos(\theta + \psi_r) \end{bmatrix} \tag{46}$$

$$\boldsymbol{G}_z = \partial g / \partial z \tag{47}$$

$$\boldsymbol{G}_z = \begin{bmatrix} \frac{\partial g_1}{\partial r} & \frac{\partial g_1}{\partial \theta} \\ \frac{\partial g_2}{\partial r} & \frac{\partial g_2}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta + \psi_r) & -r\sin(\theta + \psi_r) \\ \sin(\theta + \psi_r) & r\cos(\theta + \psi_r) \end{bmatrix} \tag{48}$$

2.5 Structure of state covariance matrix

The covariance matrix has some structure to it. It can be partitioned into map $\boldsymbol{P}_{mm}$ and the robot $\boldsymbol{P}_{rr}$ covariance blocks.

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{rr} & \boldsymbol{P}_{rm} \\ \boldsymbol{P}_{mr} & \boldsymbol{P}_{mm} \end{bmatrix} \tag{49}$$

Off diagonals $\boldsymbol{P}_{rm}$ and $\boldsymbol{P}_{mr}$ blocks are the correlations between map and robot since they are interrelated. From the moment of initialisation, the feature position is a function of the robot position hence errors in the robot position will also appear as errors in the feature position. Every observation of a feature affects the estimate of every other feature in the map (Newman 2006).

3. Consistency of EKF SLAM

SLAM is a non-linear problem hence it is necessary to check if it is consistent or not. This can be done by analysing the errors. The filter is said to be unbiased if the Expectation of the actual state estimation error, $\tilde{\boldsymbol{X}}(k)$ satisfies the following equation:

$$E\left[\tilde{\boldsymbol{X}}(k)\right] = 0 \tag{50}$$

$$E\left[\left(\tilde{\boldsymbol{X}}(k)\right)\left(\tilde{\boldsymbol{X}}(k)\right)^T\right] \leq \boldsymbol{P}(k \mid k-1) \tag{51}$$

, where the actual state estimation error is given by:

$$\tilde{\boldsymbol{X}}(k) = \boldsymbol{X}(k) - \boldsymbol{X}(k \mid k-1) \tag{52}$$

$\boldsymbol{P}(k \mid k-1)$ is the state error covariance. Equation (51) means that the actual mean square error matches the state covariances. When the ground truth solution for the state variables is

available, a chi-squared test can be applied on the normalised estimation error squared to check for filter consistency.

$$\left(\tilde{\boldsymbol{X}}(k)\right)^T \left(\boldsymbol{P}(k \mid k-1)\right)^{-1} \left(\tilde{\boldsymbol{X}}(k)\right) \leq \chi^2_{d,1-\alpha} \tag{53}$$

$d = \dim(x(k))$ and $1-\alpha$ is the desired confidence level. In most cases ground truth is not available, and consistency of the estimation is checked using only measurements that satisfy the innovation test:

$$\boldsymbol{v}_{ij}^T(k)\boldsymbol{S}_{ij}^{-1}\boldsymbol{v}_{ij}(k) < \chi^2_{d,1-\alpha} \tag{54}$$

Since the innovation term depends on the data association hypothesis, this process becomes critical in maintaining a consistent estimation of the environment map (Castellanos et al 2006).

4. Feature Extraction and Environment modelling

This is a process by which sensor data is processed to obtain well defined entities (features) which are recognisable and can be repeatedly detected. In feature based navigation methods, features must be different from the rest of the environment representation (**discrimination).** To be able to re-observe features, they must be **invariant** in scale, dimension or orientation, and they must also have a well defined **uncertainty model.** In structured domains such as indoor, features are usually modelled as geometric primitives such as points, lines and surfaces. Contrary to indoor domains, natural environments cannot be simply modelled as geometric primitives since they do not conform to any simple geometric model. A more general feature description is necessary in this regard. To aid feature recognition in these environments, more general shapes or blobs can be used and characteristics such as size, centre of mass, area, perimeter, aspects such as colour, texture, intensity and other pertinent information descriptors like mean, and variance can be extracted (Ribas 2005).

5. Data Association

In practice, features have similar properties which make them good landmarks but often make them difficult to distinguish one from the other. When this happen the problem of data association has to be addressed. Assume that at time k , an onboard sensor obtains a set of measurements $\boldsymbol{z}_i(k)$ of m environment features $\boldsymbol{E}_i(i = 1,\ldots,m)$. Data Association consists of determining the origin of each measurement, in terms of map features $F_j, j = 1,\ldots,n.$ The results is a hypothesis:

$$\boldsymbol{H}_k = \left[j_1 \quad j_2 \quad j_3 \ldots\ldots j_m \right] \tag{55}$$

, matching each measurement $\boldsymbol{z}_i(k)$ with its corresponding map feature. $F_{ji}(j_i = 0)$ indicates that the measurement $\boldsymbol{z}_i(k)$ does not come from any feature in the map (Castellanos et al. 2006). Figure 2 below summarises the data association process described here. Several techniques have been proposed to address this issue and more information on some these techniques can be found in (Castellanos et al 2006) and (Cooper 2005).
Of interest in this chapter is the simple data association problem of finding the correspondence of each measurement to a map feature. Hence the Individual Compatibility Nearest Neighbour Method will be described.

5.1 Individual Compatibility (IC)

The IC considers individual compatibility between a measurement and map feature (Castellanos et al. 2006). This idea is based on a prediction of the measurement that we would expect each map feature to generate, and a measure of the discrepancy between a predicted measurement and an actual measurement made by the sensor. The predicted measurement is then given by:

$$\boldsymbol{z}_j(k \mid k-1) = \boldsymbol{h}(\boldsymbol{X}_r(k \mid k-1), x_j, y_j) \tag{56}$$

The discrepancy between the observation $\boldsymbol{z}_i(k)$ and the predicted measurement $\boldsymbol{z}_j(k \mid k-1)$ is given by the innovation term $\boldsymbol{v}_{ij}(k)$:

$$\boldsymbol{v}_{ij}(k) = \boldsymbol{z}_i(k) - \boldsymbol{z}_j(k \mid k-1) \tag{57}$$

The covariance of the innovation term is then given as:

$$\boldsymbol{S}_{ij}(k) = \boldsymbol{H}(k)\boldsymbol{P}(k \mid k-1)\boldsymbol{H}^T(k) + \boldsymbol{R}(k) \tag{58}$$

$\boldsymbol{H}(k)$ is made up of $\boldsymbol{H}_r$, which is the Jacobian of the observation model with respect to the robot states and $\boldsymbol{H}_{Fj}$, the gradient Jacobian of the observation model with respect to the observed map feature.

$$\boldsymbol{H}(k) = \begin{bmatrix} \boldsymbol{H}_r & 0\,0 & 0\,0 & \boldsymbol{H}_{Fj} & 0\,0 \end{bmatrix} \tag{59}$$

Zeros in equation (59) above represents un-observed map features.

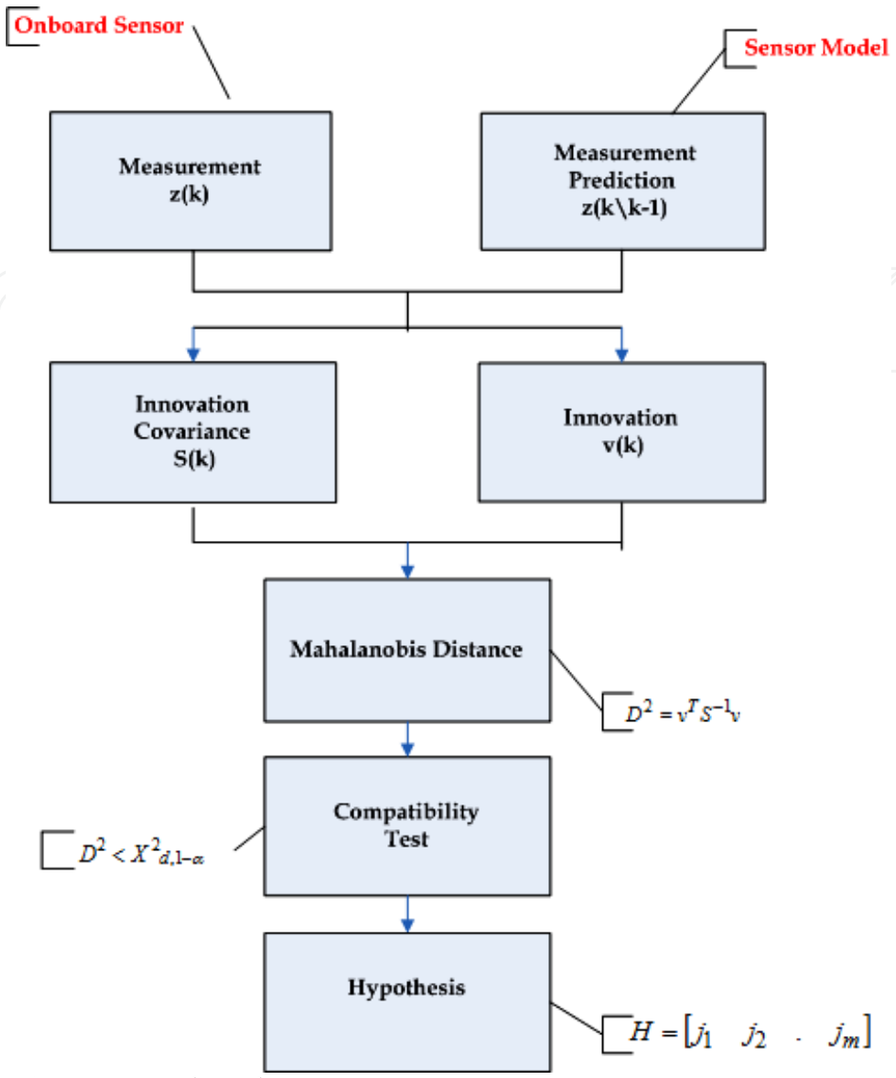


Fig. 2. Data Association Flow chart

To deduce a correspondence between a measurement and a map feature, Mahalanobis distance is used to determine compatibility, and it is given by:

$$D_{ij}^2(k) = \boldsymbol{v}_{ij}^T(k)\boldsymbol{S}_{ij}^{-1}(k)\boldsymbol{v}_{ij}(k) \tag{60}$$

The measurement and a map feature can be considered compatible if the Mahalanobis distance satisfies:

$$D_{ij}^2(k) < \chi_{d,1-\alpha}^2 \tag{61}$$

Where $d = \dim(v_{ij})$ and $1-\alpha$ is the desired level of confidence usually taken to be 95%. The result of this exercise is a subset of map features that are compatible with a particular

measurement. This is the basis of a popular data association algorithm termed Individual Compatibility Nearest Neighbour. Of the map features that satisfies IC, ICNN chooses one with the smallest Mahalanobis distance (Castellanos et al 2006).

6. Commonly used SLAM Sensors

The most popular sensor choice in indoor and outdoor applications is a laser scanner even though it is costly. Its popularity stems from the fact that it provides high quality dense data with a good angular precision. Cameras are also commonly used to obtain visual information (e.g colour, shape or texture) from the environment. Acoustic sensors are considered to be the cheapest choice but less reliable to perform SLAM even in highly structured environments. This is because sonars produce measurements with poor angular resolution and the problem of specular reflections. If used, then one must somehow deal with these limitations. The situation is different in underwater domains. Due to the attenuation and dispersion of light in water, laser based sensors and cameras become impractical, though cameras can still be used in applications where the vehicle navigates in clear water or very near to the seafloor. Due to excellent propagation of sound in water, acoustic sensors remain the best choice in the underwater domain (Ribas 2008).

7. Multi – Robot EKF – SLAM

In order for a multi-robot team to coordinate while navigating autonomously within an area, all robots must be able to determine their positions as well as map the navigation map with respect to a base frame of reference. Ideally, each robot would have direct access to measurements of its absolute position such as using GPS, but this is not possible indoor or in the vicinity of tall structures. Therefore, utilising multi robot systems becomes attractive as robots can operate individually but use information from each other to correct their estimates (Mourikis, Roumeliotis 2005).

7.1 Collaboration Vs Cooperation

There are two types of multi robot systems: collaborative and cooperative multi robot system. Collaborative case is when robots working as a team in real-time, continuously update each others state estimates with the latest sensor information. While cooperative kind is when robots share information via an external computer to find the group solution based on available communicated information (Andersson, L. 2009).
Of interest in this chapter is the Cooperative SLAM Scheme. Assuming we have two robots capable of individually performing SLAM and robot–robot relative measurements as shown in figure 3. In the picture, SLAM results and as well as possible robot-robot relative measurements results are fed into the Cooperative strategy module to calculate possible route for each member. The module utilises the current global map information to derive controls for the system. That is, determining the team manoeuvres providing optimal reward in terms of exploration gain. Figure 3 below shows a Cooperative SLAM Scheme which is an estimation and control closed loop problem similar to the structure discussed in (Andrade-Cetto,J. et al. 2005).

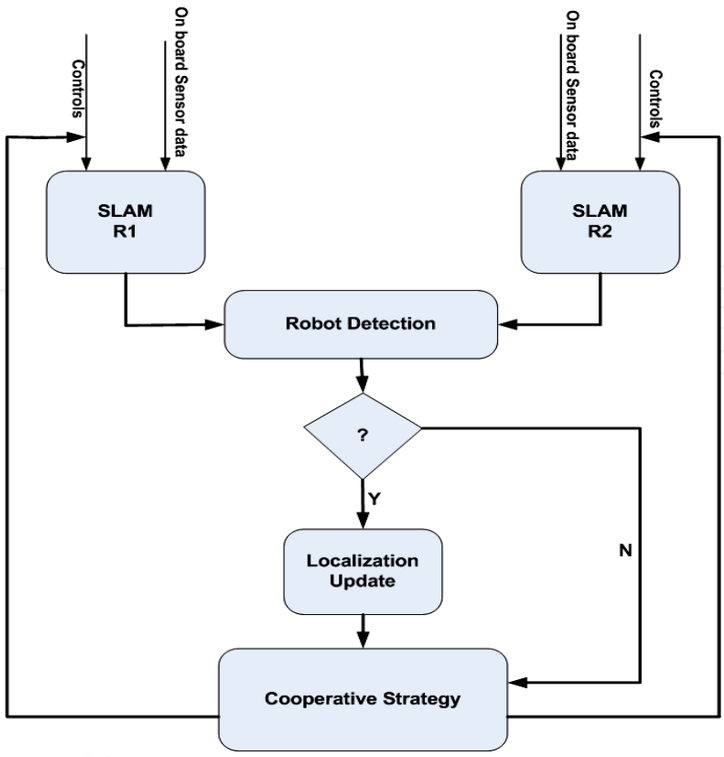


Fig. 3. Cooperative SLAM scheme

7.2 Map Initialisation

As in single robot SLAM, Multi robot SLAM system requires some common reference frame between the robots. We then make assumptions that, (1) initially the global map is empty, (2) robot one R_i starts at known pose.

$$\hat{\boldsymbol{x}}_i^B = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{62}$$

While the second robot R_j is placed in front of the first and is detectable by R_i.

7.3 State Augmentation

The first assumption states that the map is initially empty; therefore the known robots and objects need to be added to the map before updating. Using stochastic representation as discussed in (Smith, Self & Cheesman 1986), priori information can be inserted in the estimated state vector and corresponding covariance matrix as shown in equation (63 & 64) below.

$$\hat{\boldsymbol{x}}^B = \begin{bmatrix} \hat{\boldsymbol{x}}_i^B \\ \hline \hat{\boldsymbol{x}}_j^B \end{bmatrix} \tag{63}$$

$$\boldsymbol{P}^{B} = \begin{bmatrix} P(\boldsymbol{x}_i) & P(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ P(\boldsymbol{x}_j, \boldsymbol{x}_i) & P(\boldsymbol{x}_j) \end{bmatrix} \tag{64}$$

Noting that all information inserted into the state vector $\hat{\boldsymbol{x}}^{B}$ needs to be represented in the global reference frame B. From the second assumption, when a robot is observed by R_i for the first time, the system state vector needs to be augmented by the pose of the observed robot with respect to frame of the observing robot. The observed pose estimate is then transformed from the observer frame of reference to global frame through the following equations:

$$\boldsymbol{g}^{B} = \hat{\boldsymbol{x}}_j^{B} = \hat{\boldsymbol{x}}_i^{B} + R\left(\hat{z}_j^{i}\right) \tag{65}$$

$$R\left(\hat{z}_j^{i}\right) = \begin{bmatrix} r\cos(\theta_j^{i} + \psi_i) \\ r\sin(\theta_j^{i} + \psi_i) \end{bmatrix}, \tag{66}$$

where $R\left(\hat{z}_j^{i}\right)$ is a nonlinear transform function giving the spatial perception by the observer robot. The corresponding observation covariance matrix $\boldsymbol{R}$ is represented in the observer's reference frame and also needs to be transformed into global frame as shown in equation (67), i.e. $\boldsymbol{G}_z\boldsymbol{R}\boldsymbol{G}_z$. The aforementioned measurement covariance matrix $\boldsymbol{R}$ differs with sensor type, but in this chapter, we model the laser range finder type of sensor, providing range and bearing. Where range and bearing are respectively given by r and θ_j^{i} from equation (66). And ψ_i is the orientation of the observer robot.

A correct stochastic map needs the independences and interdependences to be maintained through at the mapping process. Since no absolute object (robots or landmarks) locations are given prior, the estimations of objects positioning are correlated and strongly influenced by the uncertainty in the robot(s) locations. Equation (64) is a covariance matrix of the system, where the leading diagonal elements are the variances of the state variables for R_i and R_j respectively. These are evaluated similar to the derivation in (Martinelli, Pont & Siegwart 2005), that is:

$$\boldsymbol{P}_{(k\text{-}1|k\text{-}1)}(\boldsymbol{x}_i) = \nabla \boldsymbol{f}_{k-1}{}^{i}\, \boldsymbol{P}_{(k\text{-}1|k\text{-}1)}(\boldsymbol{x}_i) \nabla \boldsymbol{f}_{k-1}{}^{iT} + \quad \boldsymbol{G}_z\boldsymbol{R}\boldsymbol{G}_z\,, \tag{67}$$

where $\nabla \boldsymbol{f}_{k-1}{}^{i}$ is the Jacobian of equation (65) with respect to the observer robot state, derived as;

$$\nabla \boldsymbol{f}_{k-1}{}^{i} = \frac{\partial \boldsymbol{g}^{B}}{\partial x_{i}{}^{B}} = \begin{bmatrix} 1 & 0 & -\rho\sin(\theta_j^i + \psi_i) \\ 0 & 1 & \rho\cos(\theta_j^i + \psi_i) \\ 0 & 0 & 1 \end{bmatrix} \tag{68}$$

G_z is also the Jacobian of equation (65) with respect to the observation. And it is calculated as:

$$\boldsymbol{G}_z = \frac{\partial \boldsymbol{g}^{B}}{\partial z_j{}^{i}} = \begin{bmatrix} \cos(\theta_j^i + \psi_i) & -r\sin(\theta_j^i + \psi_i) \\ \sin(\theta_j^i + \psi_i) & r\cos(\theta_j^i + \psi_i) \end{bmatrix} \tag{69}$$

The off diagonal elements are computed as:

$$\boldsymbol{P}(x_j, x_i) = \nabla \boldsymbol{f}_{k-1}^{j} \boldsymbol{P}(x_j, x_i) \nabla \boldsymbol{f}_{k-1}^{i\ T}, \tag{70}$$

where

$$\boldsymbol{P}(x_j, x_i) = \boldsymbol{P}(x_j, x_i)^T \tag{71}$$

The results of equations (67),(70) and (71) are then substituted back into equation (64) giving the stochastic map estimates at the initial poses.
The discrete conditional subscript used in equation (67) and the reminder of the paper is for readability purposes therefore, $\boldsymbol{P}_{k-1|k-1}$ implies $\boldsymbol{P}(k-1 \mid k-1)$.

7.4 Robot Re-observation

If a robot already in the map is re-observed, the system state vector is not augmented but it is updated. Thus, if we assume that robot R_i located at pose $\boldsymbol{x}_i^B$ makes an observation $\boldsymbol{z}_j^i$ of another robot at pose $\boldsymbol{x}_j^B$ then observation function is given as:

$$\boldsymbol{z}_j^i = \boldsymbol{h}(\boldsymbol{x}_j, \boldsymbol{x}_i) + \boldsymbol{w}_{k-1} \tag{72}$$

Assuming $\boldsymbol{w}_{k-1}$ is zero mean and of Gaussian distribution form, with covariance $\boldsymbol{R}$, calculated as:

$$\boldsymbol{R} = \begin{bmatrix} \Re_i & 0_{3\times3} \\ 0_{3\times3} & \Re_j \end{bmatrix} \tag{73}$$

Suppose the two robot are equipped with the same sensor, then $\Re_i$ and $\Re_i$ are the observation covariances for the corresponding robot. The term $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in equation (72) is a nonlinear function that relates the output of the sensor to the states. The function is made up of relative distance and relative bearing (R_i observes R_j), given as:

$$\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{bmatrix} \sqrt{(x_j - x_i)^2 \quad + \quad (y_j - y_i)^2} \\ a\tan\left(\dfrac{-\sin\psi_i(x_j - x_i) + \cos\psi_i(y_j - y_i)}{\cos\psi_i(x_j - x_i) + \sin\psi_i(y_j - y_i)}\right) \end{bmatrix} \tag{74}$$

Its jacobian is calculated as follows:

$$\nabla \boldsymbol{h} = \begin{bmatrix} \dfrac{x_j - x_i}{(x_j - x_i)^2 + (y_{r2} - y_{r1})^2} & \dfrac{y_j - y_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & 0 & \dfrac{x_j - x_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & \dfrac{y_j - y_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & 0 \\ \dfrac{y_j - y_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & \dfrac{x_j - x_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & -1 & \dfrac{y_j - y_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & \dfrac{x_j - x_i}{(x_j - x_i)^2 + (y_j - y_i)^2} & 0 \end{bmatrix} \tag{75}$$

Given prior and current measurement information we can update state estimate using Extended Kalman Filter (EKF). The filter update equation is evaluated as:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}^B_{k|k-1} + \boldsymbol{K}\left[\boldsymbol{z}^j_j - \boldsymbol{h}(\hat{\boldsymbol{x}}^B_j, \hat{\boldsymbol{x}}^B_i)\right] \tag{76}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}\nabla\boldsymbol{h}\boldsymbol{P}_{k|k-1} \tag{77}$$

$$\boldsymbol{K} = \boldsymbol{P}_{k|k-1}\nabla\boldsymbol{h}^T\left(\nabla\boldsymbol{h}\boldsymbol{P}_{k|k-1}\nabla\boldsymbol{h}^T + \boldsymbol{R}\right)^{-1} \tag{78}$$

$\hat{\boldsymbol{x}}_{k|k}$ implies $\hat{x}(k-1|k-1)$.

8. Conclusions

EKF is a good way to learn about SLAM because of its simplicity whereas probabilistic methods are complex but they handle uncertainty better. This chapter presents some of the basics feature based EKF-SLAM technique used for generating robot pose estimates together with positions of features in the robot's operating environment. It highlights some of the basics for successful EKF – SLAM implementation:, these include: Process and observation models, Basic EKF-SLAM Steps, Feature Extraction and Environment modelling, Data Association, and Multi – Robot – EKF – SLAM with more emphasis on the Cooperative

SLAM Scheme. Some of the main open challenges in SLAM include: SLAM in large environments, Large Scale Visual SLAM, Active and Action based SLAM; development of intelligent guidance strategies to maximise performance of SLAM, Autonomous Navigation and Mapping in Dynamic Environments, and 3D Mapping techniques.

9. References

Andersson, L.(2009). Multi-robot Information Fusion: Considering spatial uncertainty models, Doctoral thesis, Linköping University

Andrade-Cetto J, T. Vidal-Calleja, and A. Sanfeliu.(2005). Multirobot C-SLAM: Simultaneous localization, control, and mapping. In *Proceedings of the IEEE ICRA Workshop on Network Robot Systems*, Barcelona.

Castellanos, J.A.; Neira, J.; Tardos, J.D. (2006). Map Building and SLAM Algorithms, *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*, Lewis, F.L. & Ge, S.S. (eds), 1st edn, pp 335-371, CRC, 0-8493-3748-8, New York, USA

Cooper, A.J. (2005). A Comparison of Data Association Techniques for Simultaneous Localisation and Mapping, Masters Thesis, Massachusets Institute of Technology

Martinelli, A., Pont, F. & Siegwart, R. (2005). Mult-irobot localization using relative observations, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp 2808-2813

Mourikis, A.I. & Roumeliotis, S.I. (2005). Performance Bounds for Cooperative Simultaneous Localization and Mapping (C-SLAM), *In Proc. Robotics: Science and Systems Conference*, June 8-11, 2005, pp. 73-80, Cambridge, MA

Neira, J. (2008). The state of the art in the EKF solution to SLAM, *Presentation*

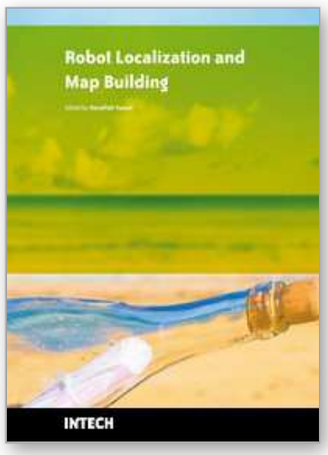
Newman, P. (2006). EKF Based Navigation and SLAM, *SLAM Summer School 2006*

Ribas, D. (2008). Underwater SLAM For Structured Environments Using and Imaging Sonar, *Phd Thesis*, University of Girona

Ribas, D. (2005). Towards Simultaneous Localization & Mapping for an AUV using an Imaging Sonar, *Masters Thesis*, University of Girona

Smith, R., Self, M. & Cheesman, P. (1986), Estimating uncertain spatial relationships in robotics, *Proceedings of the 2nd Annual Conference on Uncertainty in Artificial Intelligence, (UAI-86)*, pp. 435–461, Elsevier Science Publishing Company, Inc., New York, NY

Williams, S.B.; Newman, P.; Rosenblatt, J.; Dissanayake, G. & Whyte, H.D. (2001). Autonomous underwater navigation and control, *Robotica*, vol. 19, no. 5, pp. 481-496



Robot Localization and Map Building

Edited by Hanafiah Yussof

ISBN 978-953-7619-83-1
Hard cover, 578 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Localization and mapping are the essence of successful navigation in mobile platform technology. Localization is a fundamental task in order to achieve high levels of autonomy in robot navigation and robustness in vehicle positioning. Robot localization and mapping is commonly related to cartography, combining science, technique and computation to build a trajectory map that reality can be modelled in ways that communicate spatial information effectively. This book describes comprehensive introduction, theories and applications related to localization, positioning and map building in mobile robot and autonomous vehicle platforms. It is organized in twenty seven chapters. Each chapter is rich with different degrees of details and approaches, supported by unique and actual resources that make it possible for readers to explore and learn the up to date knowledge in robot navigation technology. Understanding the theory and principles described in this book requires a multidisciplinary background of robotics, nonlinear system, sensor network, network engineering, computer science, physics, etc.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Oduetse Matsebe, Molaletsa Namoshe and Nkgatho Tlale (2010). Basic Extended Kalman Filter – Simultaneous Localisation and Mapping, Robot Localization and Map Building, Hanafiah Yussof (Ed.), ISBN: 978-953-7619-83-1, InTech, Available from: http://www.intechopen.com/books/robot-localization-and-map-building/basic-extended-kalman-filter-simultaneous-localisation-and-mapping

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen